

# Multilingual Summary Generation in a Speech-To-Speech Translation System for Multilingual Dialogues\*

Jan Alexandersson, Peter Poller, Michael Kipp, Ralf Engel

DFKI GmbH

Stuhlsatzenhausweg 3

66123 Saarbrücken

{alexandersson,poller,engel,kipp}@dfki.de

## Abstract

This paper describes a novel functionality of the VERBMOBIL system, a large scale translation system designed for spontaneously spoken multilingual negotiation dialogues. The task is the on-demand generation of dialogue scripts and result summaries of dialogues. We focus on *summary generation* and show how the relevant data are selected from the dialogue memory and how they are packed into an appropriate abstract representation. Finally, we demonstrate how the existing generation module of VERBMOBIL was extended to produce multilingual and result summaries from these representations.

## 1 Introduction

In the last couple of years different methods for summarization have been developed. In this paper we report on a new system functionality within the scope of VERBMOBIL (Bub et al., 1997), a fully implemented speech-to-speech translation system, that generates German or English *dialogue scripts* (Alexandersson and Poller, 1998) as well as German or English *summaries* of a multilingual negotiation dialogue held with assistance of the system. By a *script* we mean a document that reflects the domain-specific propositional contents of the individual turns of a dialogue as a whole, while a *summary* gives a compact summarization of all negotiations the dialogue participants agreed on.

The key idea behind our approach is to utilize as many existing resources as possible. Conceptually we have added one module (although technically realized in different already existing modules of the overall VERBMOBIL system) – the *summary generator*. Besides formatting, our new module generates sequences of language specific (i.e., German) semantic representations for the generation of summaries/scripts based on the content of the dialogue memory (Kipp et al., 1999). These descriptions are

realized into text by the existing VERBMOBIL generator (Becker et al., 1998). To produce multilingual summaries we utilize the transfer module of VERBMOBIL (Dorna and Emele, 1996).

The next section gives an overview of the VERBMOBIL system focusing on the modules central for the production of summaries/scripts. It is followed by a section describing the extraction and maintenance of summary relevant data. We then describe the functionality of the summary generator in detail. An excerpt of the sample dialogue we refer to in the paper is given at the end of the paper.

## 2 Prerequisites

VERBMOBIL is a speech-to-speech translation project, which at present is approaching its end and in which over 100 researchers<sup>1</sup> at academic and industrial sites are developing a translation system for multilingual negotiation dialogues (held face to face or via telephone) using English, German, and Japanese. The main difference between VERBMOBIL and, c.f., man-machine dialogue systems is that VERBMOBIL mediates the dialogue instead of controlling it. Consequently, the complete dialogue structure as well as almost the complete macro-planning is out of the system's control.

The running system of today is complex, consisting of more than 75 separate modules. About one third of them concerns linguistic processing and the rest serves technical purposes. (For more information see for instance (Bub et al., 1997)). For the sake of this paper we concentrate on a small part of the system as shown in figure 1.

A user contribution is called a *turn* which is divided into *segments*. A segment ideally resembles a complete sentence as we know it from traditional grammars. However, because of the spontaneity of the user input and because the turn is chunked by a statistical process, the input segments for the linguistic components are sometimes merely pieces of linguistic material. For the dialogue memory and one of the shallow translation components the *dia-*

\* The research within VERBMOBIL presented here is funded by the German Ministry of Research and Technology under grant 01IV101K/1. The authors would like to thank Tilman Becker for comments on earlier drafts on this paper, and Stephan Lesch for invaluable help with programming.

<sup>1</sup>See <http://verbmobil.dfki.de> for the list of project partners.

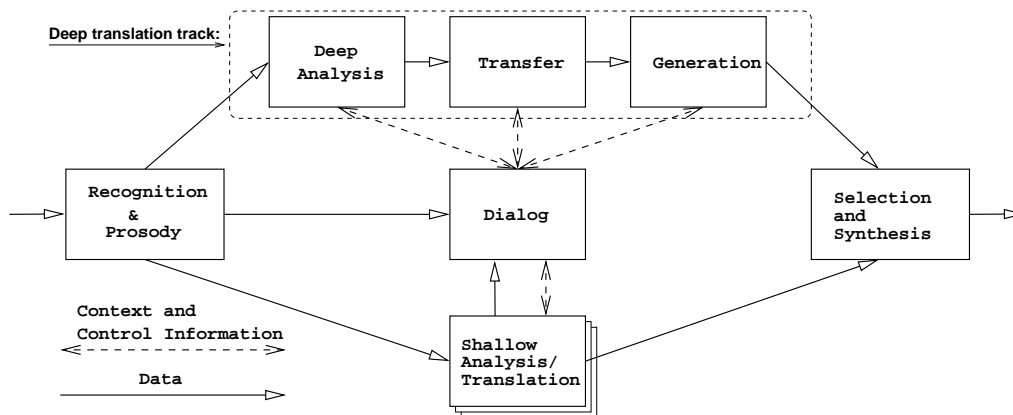


Figure 1: Part of the VERBMOBIL system

*logue act* (Alexandersson et al., 1998) plays an important role. The dialogue act represents the communicative function of an utterance, which is an important information for the translation as well as the modeling of the dialogue as a whole. Examples of illocutionary acts are REQUEST and GREET. Other acts can carry propositional content, like SUGGEST and INFORM\_FEATURE.

To obtain a good translation and enhance the robustness of the overall system the translation is based on several competing translation tracks, each based on different paradigms. The *deep translation track* consists of an HPSG based analysis, semantic transfer and finally a TAG-based generator (VM-GECO). The linguistic information within this track is encoded in a so-called VIT<sup>2</sup> (Bos et al., 1996; Dorna, 1996) which is a formalism following DRT. It consists of a set of semantic conditions (i.e. predicates, roles, operators and quantifiers) and allows for underspecification with respect to scope and subordination or inherent underspecification. A graphical representation of the VIT for the English sentence “They will meet at the station” is shown in figure 2.

Besides the deep translation track several shallow tracks have been developed. The main source of input for the generation of summaries comes from one of these shallow analysis components (described in section 3) which produces dialogue acts, topic suggestions and expressions in a new knowledge representation language called DIREX<sup>3</sup>. These expressions represent domain related information like source and destination cities, dates, important hotel related data, and meeting points. This input is processed by the dialogue module which computes the relevant (accepted) *objects* of the negotiation (each consisting of dialogue act, topic, and a DIREX)

Figure 3 shows the conceptual architecture, where

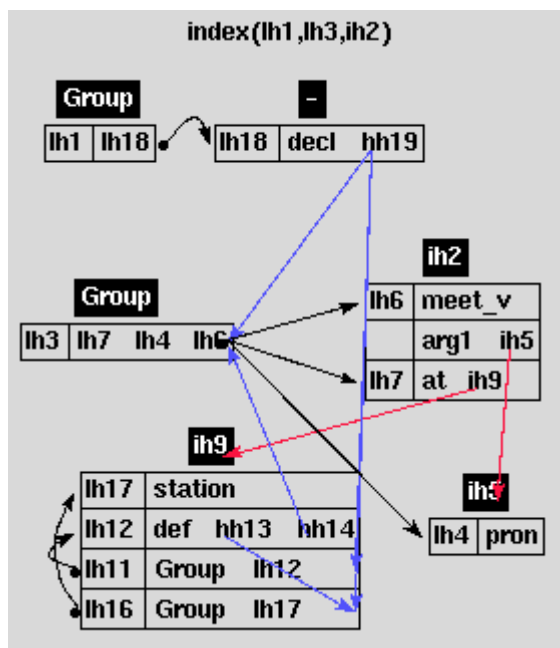


Figure 2: Graphical representation of VIT for “They will meet at the station”

the summary generation process as a whole is indicated with thicker lines. It consists of the following steps:

- **Content Selection:** The relevant structures are selected from the dialogue memory.
- **Summary Generation:** These structures are converted into sequences of semantic descriptions (VITs) of full sentences for *German* (see section 4).
- **Transfer:** Depending on the target language, the German sentence VITs are sent through the transfer module.
- **Sentence Generation:** The VITs are generated by the existing VERBMOBIL generator (Becker et al.,

<sup>2</sup>Verbmobil Interface Term

<sup>3</sup>Domain Representation Expression

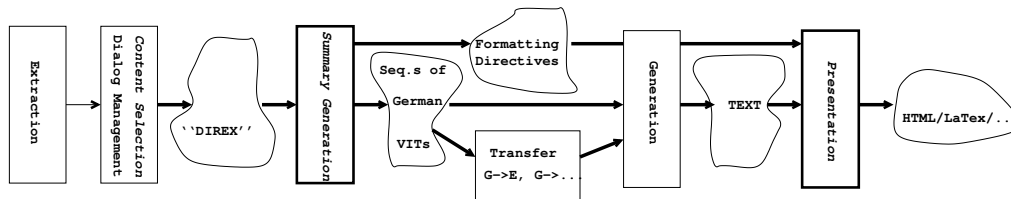


Figure 3: Conceptual Architecture of the Summary Generation Process

2000).

- **Presentation:** The sentences are incorporated into the final, e.g., HTML document.

Throughout the paper we will refer to a German-English dialogue (see appendix for an excerpt). The information presented there is the spoken sentence(s) together with the information extracted as described in section 3. To save space we only present parts of it, namely those which give rise to the structures in figure 4.

### 3 Extraction and Maintenance of Protocol Relevant Data

The dialogue memory gets its input from one of the shallow translation components, which bases its translation on the dialogue act and DIREX-expression extracted from the segment. The input is a triple consisting of:

- **Dialogue Act** representing the intention of the segment.
- **Topic** is one of the four topics *scheduling, traveling, accommodation* and *entertainment*.
- **Direx** representing the propositional content of the segment.

For the extraction of propositional content and intention we use a combination of knowledge based and statistical methods. To compute the propositional content finite state transducers (FSTs) (Appelt et al., 1993) with built-in functions are used (Kipp et al., 1999). The intention (represented by a dialogue act) is computed statistically using language models (Reithinger and Klesen, 1997). Both methods were chosen because of their robustness – since the speech recognizers have a word error rate of about 20%, we cannot expect sound input for the analysis. Also the segmentation of turns in utterances is stochastic and therefore sometimes delivers suboptimal segments. Consider the input to be processed:

I would so we were to leave Hamburg on the first

where the speech recognizer replaced “good so we will” with “I would so we were to”. The result of the extraction module looks like:

```
[INFORM,traveling,has_move:[move,
has_source_location:[city,has_name=
'hamburg'],has_departure_time:
[date,time=[day:1]]]]
```

The result consists of the dialogue act INFORM, the topic suggestion *traveling*, and and a DIREX. The top object is a move with two *roles*: A source location (which is a city – Hanover), and a departure time (which is a date – day 1).

#### Dialog processing

For each utterance, and hence each DIREX the dialogue manager (1) estimates its relevance, and (2) enriches it with context. For summary generation, we are solely interested in the *most specific, accepted* objects. Therefore, we also (3) compute *more specific/general* relations between objects:

**Relevance detection.** Depending on the dialogue act of the current utterance different courses of action are taken. SUGGEST dialogue acts trigger the storage, completion, focusing and inter-object relation (see below) computation for the current structure. ACCEPT and REJECT acts let the system mark the focused object accepted/rejected.

**Object Completion.** Suggestions in negotiation dialogues are incomplete most of the time. E.g., the utterance “I would prefer to leave at five” is a suggestion referring to the departure time for a trip from Munich to Hanover on the 19. Jan. 2000 (see turn 1005 in the appendix). Most of the complete data has been mentioned in the preceding dialogue. Our completion algorithm uses the focused object (itself a completed suggestion) to complete the current structure. All non-conflicting information of the focused object is *copied* onto the new object. In our example the current temporal information “I would prefer to leave at five” would be completed with date (i.e., “19. Jan. 2000”) and other travel data (“trip from Munich to Hanover”). Afterwards, it will be put to focus.

**Object Relations.** The processing results in a number of accepted and rejected objects. Normally, a negotiation produces a series of suggestions that become more specific over time. For each new object we calculate the *relation* to all other suggestions in terms of *more specific/general* or *equal*. A final inference

procedure then filters redundant objects and produces a list of accepted objects with highest specificity. Figure 4 shows two such objects extracted from the sample dialogue. Both structures have been completed from context data including situational data, i.e., current time and place of the negotiation.

```

=====
Topic SCHEDULING
=====

relations:
((MORE_SPECIFIC_THAN . #<APPOINTMENT P2*>))

APPOINTMENT (P5**0)
  HAS_LOCATION --> CITY (P4*)
    HAS_NAME="hannover"
  HAS_MEETING --> MEETING (P3**)
    HAS_NAME="geschaefststreffen"
  HAS_DATE --> DATE (P5*)
    TEMPEX=[year:2000,
            month:jan,
            day:20,
            part:am,
            time:11:0]

relations:
((MORE_SPECIFIC_THAN . #<APPOINTMENT P26*>
 (MORE_SPECIFIC_THAN . #<APPOINTMENT P30**+0>))

APPOINTMENT (P29**0)
  HAS_LOCATION --> NONGEO_LOCATION (P30***)
    HAS_NAME="bahnhof"
  HAS_DATE --> DATE (P29*)
    TEMPEX=[year:2000,
            month:jan,
            day:19,
            time:9:30]

```

Figure 4: The scheduling part of the thematic structure

## 4 Generating Summaries

Our system uses many of the existing components of VERBMOBIL. However, we had to develop a new component, the *summary generator*, which is described below. It solves the task of mapping the DIREX structures selected in the dialogue memory into sequences of full fledged semantic sentence descriptions (VITs), thereby performing the following steps:

- **Document Planning:** Extracting, preparing and dividing the content of the dialogue memory into a predefined format. This includes, c.f., time/place of negotiation, participants, result of the negotiation.
- **Sentence Planning:** Splitting the input into chunks suitable for a sentence. This process involves choosing an appropriate verb and arranging the parts of the chunk as arguments and/or adjuncts. The final step is the mapping of this internal

representation onto a semantic description (VIT) for each sentence (suitable for further processing by the existing VERBMOBIL components).

- **Generation:** Verbalizing the VITs by the existing multilingual generator of VERBMOBIL.
- **Presentation:** Formatting of the complete document content to an, e.g., HTML-page. Finally, the document is displayed by an appropriate browser.

Our approach has been mostly guided by robustness: our representation language (DIREX) was co-developed during the course of the project. Moreover, as the extraction component increased its vocabulary, we wanted to be able to generate new information which had not been seen before. Hence we needed an approach which is fault tolerant. Instead of failing when the representation changes or new type of objects were introduced we degrade in precision. Our two step approach has proven its usefulness for this.

### 4.1 Document Planning

The document itself contains two main parts. The top of the document includes general information about the dialogue (place, date, participants, theme). The body of the document contains the summary part which is divided into four paragraphs, each of them verbalizing the agreements for one negotiation topic: *scheduling*, *accommodation*, *traveling* and *entertainment*. Therefore, our document planning is very straightforward. The four elements of the top document are processed in the following manner:

- *Place and Date:* For place and date the information is simply retrieved from the dialogue memory.
- *Participants:* The participants information are transformed into a VIT by the *plan processor* described below. In the absence of name/title information, a character, e.g., A, B, ... is used.
- *Theme:* By a shallow examination of the result of the content extraction, a semantic description corresponding to a noun phrase mirroring the content of the document as a whole is construed. An example is *Business trip with accommodation*.
- *The summary:* Finally, the summary relevant DIREX objects are retrieved from the dialogue memory: First we compute the most specific suggestions by using the *most specific/general* and *equal* relations. The remaining suggestions are partitioned into equivalence classes which are filtered by computing the degree of acceptance. In case of conflict the most recent one is taken. The resulting set is partitioned into the above mentioned topics they belong to. Finally these are processed by the plan processor as described below.

### 4.2 Sentence Planning

We now turn into the process of mapping the interesting part of the dialogue memory onto sequences

of VITs. An example of the content of one topic – scheduling – was shown in figure 4. Our two step approach consists of:

- **A plan processor** whose task it is to split the objects selected into chunks suitable for a sentence. Possibly it contributes to the selection of verbs.
- **A semantic constructor** whose task it is to convert the output of the plan processor into full fledged semantic descriptions (VITs) for the sentences of the document. This second step can be viewed as a robust fall-back: If the plan processor does not succeed in obtaining full specifications of all sentence parts, this step secures a valid and complete specification.

#### 4.2.1 The plan processor

Input to the plan processor (Alexandersson and Reithinger, 1997) is the thematic structure partly shown in figure 4. The plan processor interprets (currently about 150) plan operators which are expanded in a top-down left to right fashion.

For the overall structure of the text, the imposed topic structure of the thematic structure is kept. Within a topic we use a set of operators which are capable of realizing (parts of) the structures to NPs, PPs and possibly verb information forming a high level specification of a sentence.

#### Plan operators

A plan operator consists of a goal which is optionally divided into subgoal(s). Its syntax contains the keywords `:constraints` and `:actions` which can be any Lisp expression. Variables are indicated with question/exclamation marks (see figures 5 and 6).

The goal of the operators uses an interface based on a triple with the following usage:

- `<description>` This is the input position of the operator. It describes and binds the object which will be processed by this operator.
- `<context>` This is the context – input/output. The context contains a stack for objects in focus, handled as described in (Grosz and Sidner, 1986). Additionally we put the generated information on a history list (Dale, 1995). The context supports the generation of, e.g., pronouns (see below). At present the context is only used local to each topic.
- `<output>` The result of the operator. The possible output types are NP, PP and sentence(s).

We distinguish two types of operators; *complex* operators, responsible for complex objects, which can contain several roles, and *simple* operators, which can process simple objects (carrying only one role). The general design of a complex operator – see figure 5 for an operator responsible for appointment objects – consists of three subgoals:

- `(find-roles ...)` Retrieve the content of the object. The operators responsible for solving the `find-roles` goal optionally allow for an enumeration of the roles we want to use.

- `(split-roles ...)` These roles (and values) will be partitioned into chunks (which we call a *split*) suitable for generating one sentence.
- `(generate-splits ...)` Finally the output – a sentence description – will be constructed.

```
(defplan appointment
  :goal ((class (?app scheduling))
         (?in-context ?out-context)
         ?sentence)
  :constraints (appointment-p !app)
  :subgoals (:sequence
            (find-roles ?app ?rels)
            (split-roles ?rels
              appointment ?1-of-splits)
            (generate-splits ?1-of-splits
              (?in-context ?out-context)
              appointment ?sentence)))
```

Figure 5: An example of an operator for a “complex” object

Behind the functionality of the `split-roles` goal we use pairs of operators (figure 6), where the first is a fact describing the roles of the split, and the second is a description for how to realize the sentence. In this example the selection of an appropriate verb is not performed by this operator but by the semantic constructor.

The second type of operators are simple operators like the one for the generation of time expressions (`tempex`) or cities (see figure 4).

Figure 7 shows a simplified plan processor output (building block) for one sentence.

#### 4.2.2 The Semantic Constructor

The task of the semantic constructor is to map the information about sentences computed by the plan processor to full semantic representations (VITs).

The knowledge source for this computational step is a declarative set of about 160 different semantically oriented sentence patterns which are encoded in an easily extendable semantic/syntactic description language.

To obtain a complete semantic representation for a sentence we first select a sentence pattern. This pattern is then, together with the output of the plan processor, interpreted to produce the VIT. The selection criteria for a sentence pattern are:

- All patterns are ordered topic-wise because the appropriateness of sentence patterns is topic-dependent (e.g., the insertion of topic-specific NPs or PPs into a sentence).
- The intentional state of the information to be verbalized highly restricts the set of appropriate verbs.
- Depending on the propositional content described within a DIREX-VIT – i.e., a VIT representing one sentence part in a building block of the

```

;; - Das <Treffen> findet in <City>
;; am <tempex> statt
;; - The <Meeting> takes place
;; in <City> on the <tempex>
(deffact sentence-split
  :goal (sentence-split
        ((has_meeting ?has_name)
         (has_location ?has_location)
         (has_date ?has_date))
        ?_topic))
(defplan generate-split
  :goal (generate-split
        ((has_meeting ?name) ;; meeting
         (has_location ?location) ;; city
         (has_date ?date) ;; tempex
         (?in-context ?out-context)
         ?topic
         ?s)
        :subgoals
        (:seq ((class (?location ?scheduling pp))
              ?topic ?loc-pp)
              ((class (?name ?scheduling))
               ?topic ?s-topic)
              (generate-full-tempex ?date ?tempex)
              ((generate-sentence decl)
               (subj ?topic has_topic)
               (obj ?l-pp has_location)
               (obj-add ?tempex has_date))
              ?in-context ?out-context ?s)))

```

Figure 6: Example of sentence definition and generation

```

(ACCOMMODATION
 (ACCEPTED
  (HAS_SIZE VIT:<Einzelzimmer>)
  (HAS_PRICE VIT:<80-Euro-pro-Nacht>))
))

```

Figure 7: Example of a plan processor output

plan processor output – it has to play different semantic roles in the sentence (e.g., verb–argument vs. verb–complement)

→ Additionally, the number of DIREX–VITs given within a building block for a sentence, influences the distribution of them to appropriate semantic roles.

Figure 8 shows a simplified sentence pattern that is selected for the building block in figure 7 to construct a VIT for, e.g., the German sentence *Das Einzelzimmer kostet 80 Euro pro Nacht*. (“The single room costs 80 euro per night.”). According

```

((:verb kosten_v)
 (:subj HAS_SIZE)
 (:obj HAS_PRICE)
 (:rest DIREX_PPS))

```

Figure 8: Example of a sentence pattern

to the above mentioned selection criteria, this pat-

tern is selected only for building blocks within the accommodation topic that contain at least values for the roles HAS\_SIZE and HAS\_PRICE, respectively. The sentence pattern contains the following “building instructions”: The semantic verb predicate (:verb) is *kosten\_v* (to cost), its subject argument (:subj) is to be filled by the DIREX–VIT associated to the DIREX–role HAS\_SIZE while :obj means a similar instruction for the direct object. The robustness fallback (:rest DIREX\_PPS) means that all other DIREX–VITs are attached to the verb as PP complements. It is part of all sentence patterns to ensure that even erroneous building blocks or erroneously selected sentence patterns produce a sentence VIT.

Finally, the VIT is constructed by interpreting the sentence pattern. The interpreter walks through the sentence pattern and performs different actions depending on the keywords, e.g., :verb, :subj and their values.

#### 4.2.3 Utilizing Context

During the course of the generation, the plan processor incrementally constructs a context (Dale, 1995), which allows for the generation of, c.f., anaphora or demonstratives for making the text fluent or contrasting purposes.

- **Anaphora** If, e.g., a meeting is split into more than one sentence, the plan processor uses an anaphora to the meeting in the second sentence.
- **Discourse Markers** In case of multiple, e.g., meetings we introduce the second with a discourse marker, e.g., “also”.
- **Demonstratives** In case of multiple meetings, we use a demonstrative to refer to the second meeting.

In addition to the plan processor, the semantic constructor also takes care of coherence within the paragraphs produced for the individual topics hereby focusing on the generation of anaphora and adverbial discourse markers. While the local context of the plan processor is based on the propositional content at hand, the semantic constructor uses a postprocessing module that is based on the output VITs of the plan processor (DIREX–VITs) using its own semantically oriented local context memory.

Anaphorization and insertion of discourse markers within the semantic constructor are based on a comparison of plan processor output VITs occurring within consecutive sentences of a paragraph. Identical verb arguments (NPs) in consecutive sentences are replaced by appropriate anaphoric pronouns while identical verbs themselves lead to the insertion of an appropriate adverbial discourse marker.

## 5 Multilinguality

The generation of dialogue scripts and result summaries is fully implemented in VERBMobil for German and English. For the English summaries we

make use of the transfer component as follows. All VITs from the German document representation are extracted, then the transfer module produces equivalent English VITs which are finally sent to the English generation component for producing the English text.

Figure 9 shows the English result summary of the dialogue shown in the appendix.

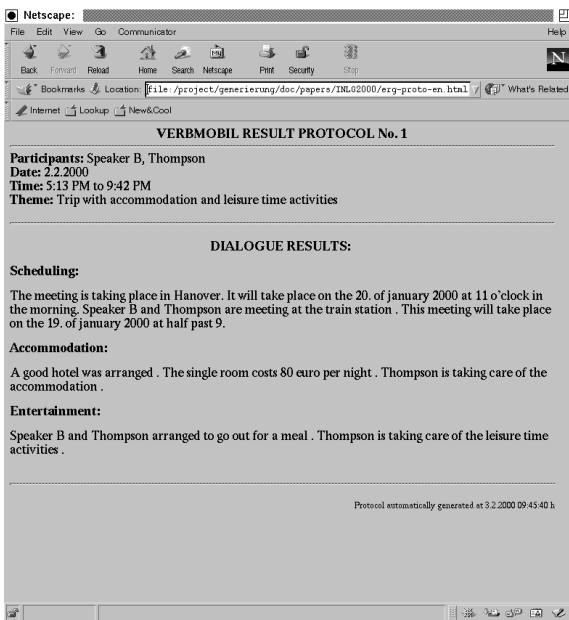


Figure 9: Example of an English result summary

## 6 Evaluation

We have performed a small evaluation of the overall system as described in this paper. Basis for the evaluation were the transcripts of four German-English negotiation dialogues. For each dialogue the resulting features of the negotiation (maximally 47, e.g., location, date for a meeting, speakers name and title, book agent) were annotated by a human, and then compared with the result of running the dialogues through the system and generating the summaries.

The features in the summary were compared using the following classifications:

- **Corr** The feature approximately corresponds to the human annotation. This means that the feature is either (1) a 100% match, (2) it was not sufficiently specified or (3) too specific. An example of (2) is when the correct date included a time, which was not captured. An example of (3) is when a date with time was annotated but the feature contained just a date.
- **Miss** A feature is not included in the summary.
- **False** A feature was erroneously included in the summary, meaning that the feature was not part of the dialogue or it received a wrong value.

- **TN** A feature was not part of the dialogue, and not included in the summary.

The evaluation result is shown in figure 10. It uses the standard precision, recall and fallout as defined in (Mani et al., 1998).

Dialogue	1	2	3	4	aver
Turns	33	33	31	32	32.25
Corr	6	13	9	11	9.75
Miss	6	3	5	4	4.5
False	3	3	3	0	2.25
TN	32	28	30	32	30.5
Recall	0.50	0.81	0.64	0.73	0.67
Prec.	0.67	0.81	0.75	1.0	0.81
Fallout	0.09	0.10	0.09	0.00	0.07

Figure 10: Evaluation Results

Obviously, our approach tries to be on the safe side; the summary contains only those features that the system thinks both partners agreed on. The main reasons for not getting higher numbers is twofold. The recognition of dialogue acts, and thus the recognition of the intension behind the utterances reaches a 70% recall (Reithinger and Klesen, 1997). We also still make errors during the content extraction.

## 7 Conclusion

We have presented an extension to existing modules allowing for the generation of summaries within the VERBMOBIL system. To our knowledge our system is the only one that uses semantic representation as basis for summarizing. Other approaches use, e.g., statistical techniques or rhetorical parsing (Waibel et al., 1998; Hovy and Marcu, 1998) to obtain the summaries. Moreover, although our module is restricted to language specific processing, the use of semantics and the transfer module allow for the generation of multilingual documents in a very straightforward fashion.

In the near future we will extend the system with respect to:

- **Sentence Split** At present the first found sentence split is chosen. This is not necessarily the optimal one. We are currently in the process of developing criteria for ranking competing results.
- **Japanese** The VERBMOBIL system currently includes German, English and Japanese. We intend to apply the same technique as for the English summaries to generate Japanese ones.

## References

- J. Alexandersson and P. Poller. 1998. Towards multilingual protocol generation for spontaneous speech dialogues. In *Proceedings of INLG-98*, Niagara-On-The-Lake, Ontario, Canada.

- J. Alexandersson and N. Reithinger. 1997. Learning dialogue structures from a corpus. In *Proceedings of EuroSpeech-97*, pages 2231–2235, Rhodes.
- Jan Alexandersson, Bianka Buschbeck-Wolf, Tsutomu Fujinami, Michael Kipp, Stephan Koch, Elisabeth Maier, Norbert Reithinger, Birte Schmitz, and Melanie Siegel. 1998. Dialogue Acts in VERBMOBIL-2 – Second Edition. Verbmobil-Report 226, DFKI Saarbrücken, Universität Stuttgart, Technische Universität Berlin, Universität des Saarlandes.
- D. Appelt, J. Hobbs, J. Bear, and M. Tyson. 1993. FASTUS: A finite-state processor for information extraction from real-world text. In *IJCAI-93*.
- T. Becker, W. Finkler, A. Kilger, and P. Poller. 1998. An efficient kernel for multilingual generation in speech-to-speech dialogue translation. In *Proceedings of COLING/ACL-98*, Montreal, Quebec, Canada.
- T. Becker, A. Kilger, P. Lopez, and P. Poller. 2000. Multilingual generation for translation in speech-to-speech dialogues and its realization in verbmobil. In *Proceedings of ECAI-2000*, Berlin, Germany.
- J. Bos, B. Gambäck, C. Lieske, Y. Mori, M. Pinkal, and K. Worm. 1996. Compositional semantics in verbmobil. In *Proceedings of Coling '96*, Copenhagen, Denmark.
- T. Bub, W. Wahlster, and A. Waibel. 1997. Verbmobil: The combination of deep and shallow processing for spontaneous speech translation. In *Proceedings of ICASSP-97*, pages 71–74, Munich.
- R. Dale. 1995. An introduction to natural language generation. Technical report, Microsoft Research Institute (MRI), Macquarie University. Presented at the 1995 European Summer School on Logic, Language and Information, Available from <http://www.mri.mq.edu.au/~rdale/nlg-textbook/ESSLI95/>.
- M. Dorna and M. Emele. 1996. Efficient Implementation of a Semantic-Based Transfer Approach. In *Proceedings of ECAI-96*, pages 567–571, Budapest, Hungary, August.
- M. Dorna. 1996. The ADT-Package for the VERBMOBIL Interface Term. Verbmobil Report 104, IMS, Universität Stuttgart, Germany.
- B. Grosz and C. Sidner. 1986. Attention, Intentions and the Structure of Discourse. *Journal of Computational Linguistics*, 12(3).
- E. Hovy and D. Marcu. 1998. Coling/acl-98 tutorial on automated text summarization. Available from <http://www.isi.edu/~marcu/coling-acl98-tutorial.html>.
- M. Kipp, J. Alexandersson, and N. Reithinger. 1999. Understanding Spontaneous Negotiation Dialogue. In *Workshop Proceedings 'Knowledge And Reasoning in Practical Dialogue Systems' of IJCAI '99*, pages 57–64.
- I. Mani, D. House, G. Klein, L. Hirschman, L. Obrist, T. Firmin, M. Chrzanowski, and B. Sundheim. 1998. The tipster summac text summarization evaluation - final report. Technical report, The Mitre Corp. Available from [http://www-24.nist.gov/related\\_projects/tipster\\_summac/final\\_rpt.html](http://www-24.nist.gov/related_projects/tipster_summac/final_rpt.html).
- N. Reithinger and M. Klesen. 1997. Dialogue Act Classification Using Language Models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, Rhodes.
- A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. 1998. Meeting Browser: Tracking and Summarizing Meetings. In *Proceedings of the DARPA Broadcast News Workshop*.
- ja es geht um das Geschäftstreffen in Hannover (lit.: Yes it is about the business meeting in Hanover)  
 [INIT,scheduling,has\_appointment:  
 [appointment,has\_meeting:[meeting,  
 has\_name='geschaefsttreffen'],  
 has\_location:[city,has\_name='hannover',  
 has\_loc\_spec=in,has\_det=unknown]]]
- das ist ja am zwanzigsten Januar um elf Uhr vormittags  
 [SUGGEST,uncertain\_scheduling,has\_date:  
 [date,tempex='(ge\_2920\_0,[from:  
 [dom:20,month:jan,tod:11:0,  
 pod:morning\_ger2]])']]
- 1003
- so we have to leave Munich at six o'clock  
 [SUGGEST,traveling,has\_move:[move,  
 has\_source\_location:[city,has\_name  
 ='muenchen'],has\_departure\_time:[date,  
 tempex='(en\_2920\_0,[from:tod:6:0])']]]]
- 1004
- vielleicht fahren wir lieber den Tag davor (lit.: maybe we better leave the day before)  
 [SUGGEST,traveling,has\_move:[move,  
 has\_departure\_time:[date,tempex=  
 '(ge\_2920\_1,[from:  
 neg\_shift(dur(1,days),ana\_point)])']]]]
- da gibt es einen Zug um zwei Uhr (lit.: there is a train at two o'clock)  
 [SUGGEST,traveling,has\_move:[move,has-  
 \_transportation:[rail],has\_departure\_time:  
 [date,tempex='(ge\_2920\_2,[from:tod:2:0])']]]]
- 1005
- I would prefer to leave at five  
 [SUGGEST,traveling,has\_move:[move,  
 has\_agent:[speaker],has\_departure\_time:  
 [date,tempex='(en\_2920\_1,[from:tod:5:0])']]]]
- [...]
- 1011
- let us meet at the station on Wednesday  
 [SUGGEST,scheduling,has\_appointment:  
 [appointment,has\_location:[nongeo\_location,  
 has\_name='bahnhof',has\_loc\_spec=at,  
 has\_det=def],has\_date:[date,tempex=  
 '(en\_2920\_2,[from:dow:wed])']]]]
- 1012
- um halb zehn am Bahnhof (lit.: at half past nine at the station)  
 [ACCEPT,uncertain\_scheduling,has\_date:[date,  
 tempex='(ge\_2920\_3,[from:tod:9:30])'],  
 has\_location:[nongeo\_location,has\_name=  
 'bahnhof']]
- [...]

## Appendix

Excerpt from our sample dialogue.

[...]  
 1002